Editor: Nargess Memarsadeghi, nargess.memarsadeghi@nasa.gov

NASA Computational Case Study: Golomb Rulers and Their Applications

Nargess Memarsadeghi | NASA Goddard Space Flight Center

Future NASA space-based astrophysics missions require high angular resolution imagery at wavelengths spanning the ultraviolet, visible, and infrared spectrums. However, achieving the required resolution is neither practical nor cost-effective with a single aperture. For example, a resolution similar to that of the Hubble space telescope's in the far infrared requires a single aperture telescope on the order of 1 km in diameter.[1] In contrast, multiple apertures combined with interferometry techniques[2] enable high-resolution data in a cost-effective manner. Similarly, in Earth sciences, multiple small radar apertures or a moving aperture can produce the effect of one large antenna, synthetically.[3]

Finding the optimal number of apertures and geometric configurations for the required spatial and spectral resolution in the minimal collecting area is of paramount importance—it drives telescope mass, volume, and complexity, and hence has a large effect on mission costs. This is the motivation behind this case study. The desired location of such apertures and antennas to maximize the return of nonredundant information is directly related to marks on *Golomb rulers*, or *Golomb rectangles*, mathematical concepts that we'll learn about here. Radio astronomers have a long history of using such nonredundant aperture patterns.[4–6]

Golomb Rulers

Let's consider an office ruler that has marks at locations 0, 1, 2, ..., 30 centimeters.

Question 1

What is the length of your ruler? How many (major) marks does it have? What are the pairwise distances of these major marks on your ruler? How many ways can you measure the same distance (for example, distances of 1-unit, 2-units, ..., L-units length)?

A typical ruler is marked at equal distances of a unit length (every inch or centimeter), and many of these marks share the same pairwise distances among them—pairwise distances of all consecutive marks are a unit long, pairwise distances of every other mark are two units long, and so on. Therefore, to measure a distance $d < l$, you can use several different pairs of marks on the ruler. To measure a 4-cm interval using a 30-cm ruler marked at 1-cm intervals, for instance, you have several options—you can use the distance between marks labeled 0 and 4, 1 and 5, 10 and 14, and so on.

Question 2

Consider a ruler of 11 units long that's marked at locations 0, 1, 4, 9, and 11. What are the pairwise distances between its marks? How many different ways can you measure distances of 1-unit, 2-units,..., 11-units long using this ruler?

The ruler you worked with in Question 2 is an example of a Golomb ruler.[7] A Golomb ruler's main property is that the distance between any pair of its marks is distinct.[8] In other words, if you can measure an integer distance of d units with this ruler, there should be only one way of making this measurement with your ruler—if indeed that ruler is a Golomb ruler.

More formally, a Golomb ruler consists of a set of integers $A = \{a_1, a_2, \ldots, a_n\}$, where $a_1 < a_2 < \ldots < a_n$, such that for each nonzero integer x, there is at most one solution to the equation $x = a_j - a_i$, $a_i, a_j \in A$. The set of integers, A, represents positions of n marks on a ruler with integer length $L = a_n - a_1$. For this case study, we consider $a_1 = 0$ and $L = a_n$.

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP November/December 2016

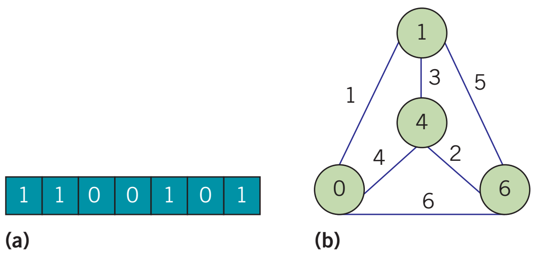

Figure 1. Two representations of a Golomb ruler of length 6 with marks at locations 0, 1, 4, and 6: (a) A one-dimensional array and (b) graph.

Question 3

What is the number of pairwise distances among marks of a ruler with n marks?

There are different ways to represent a Golomb ruler. One is to use a length $l + 1$ array of zeros and ones, with cells corresponding to marks containing value one and other cells storing zero. Another representation would be a graph whose nodes are labeled with distinct nonnegative integers (ruler marks) and whose edges are labeled with the pairwise distances of their connecting node values. Figure 1 displays these two representations for a ruler of length 6 with marks at positions {0, 1, 4, 6}. The Golomb ruler is also directly related to the concept of the Sidon sets, which are sets of natural numbers with distinct pairwise sums.[9]

The remainder of this case study considers the array representation of Golomb rulers, with cells at the mark locations represented with value one and zero otherwise. The Golomb ruler problem has variants such as Golomb rectangles,[10,11] Costas arrays,[12,13] and Honeycomb arrays[14] in 2D space, but we focus only on Golomb rulers here.

Computational Complexity and Construction Algorithms

A Golomb ruler is optimal if no shorter Golomb ruler with the same number of marks can be formed. Furthermore, if all integer distances {1, 2, ..., l} can be measured with a Golomb ruler of length l, you have a perfect Golomb ruler. The graph representation of a perfect Golomb ruler is a *graceful graph*,[7] one with n nodes such that each node is labeled with distinct nonnegative numbers no larger than n in a way that each edge is assigned exactly one of the integers from 1 to n.

What Is an NP-Hard Problem?

Suppose you are given a problem, such that its solution is a Boolean value: yes or no (or true or false). Such a problem is called a *decision problem*. There are different classes of decision problems; the set that can be solved in *polynomial time* (time that's a polynomial function of the size of the input) is called *P*. There are other decision problems that we might or might not be able to solve in polynomial time, but given an answer, we can verify its correctness in polynomial time. These problems belong to a class of problems called NP. It's easy to see P is a subset of NP. A problem is NP-hard if an algorithm for solving it could be translated in polynomial time into a method for solving any other NP-problem. An NP-hard problem may or may not be verifiable in polynomial time, and it has no known polynomial time algorithm for solving it. (See http://mathworld.wolfram.com/NP-Problem.html for more.)

Question 4

What are the distances that a perfect Golomb ruler covers in terms of n, the number of marks on the ruler? What is the length of the ruler, l, in terms of n?

Question 5

Consider the Golomb ruler in Question 2. Is it a perfect Golomb ruler? How about an optimal one? Why or why not for each case? How about the ruler in Figure 1?

Question 6

Prove Theorem 1: there is no perfect Golomb ruler with more than four marks.[7]

Hint 1: Suppose there's a perfect ruler of length L with $n > 4$ marks.

Hint 2: How can you mark the ruler so that distance of $L - 1$ is measured? How many marks does your ruler have so far?

Hint 3: Repeat the above exercise for distances of $L - 2$ and $L - 3$.

Now that we know there's no perfect Golomb ruler with more than four marks, the goal is to find optimal rulers. Finding optimal Golomb rulers is conjectured to be an *NP-hard problem* (see the sidebar), although there's no formal proof for it. There has been some work into finding the lower bounds of the solutions to this problem.[15,16] The next question helps us better understand the computational complexity involved.

Question 7

Consider a ruler of length l. How many different ways can we mark this ruler at integer positions? How many different rulers of maximum length l will have exactly n marks? How many distances should be calculated for each such ruler to check for nonredundancy of measurements? Consider a Golomb ruler of maximum length 30 with 12 marks. How long does it take to exhaustively search for the shortest of such Golomb rulers, if each measurement and test takes only one nanosecond (10^{-9} seconds)? If needed, convert the total search time to minutes, hours, or larger units of time.

The computational time for finding optimal Golomb rulers grows exponentially—for example, nonheuristic exhaustive search time for a Golomb ruler with 16 marks and of length 49 is 3.92×10^9 years.[17] Therefore, construction algorithms that generate rulers based on properties of prime numbers are of great value. Even if these construction algorithms don't provide an optimal solution, they can serve as good initial guesses to our optimization problem. For many applications, having such nonredundant patterns suffice. One of these construction algorithms is the Erdös-Turán construction:[18,19] for every odd prime number p, and for every $k \in \{0, 1, ..., p - 1\}$ the sequence formed by $2pk + (k^2 \bmod p)$ is a Golomb ruler. Other construction algorithms include Golomb, Lempel, Ruzsa, and Singer, which you can learn about.[8,12,20]

Question 8

Consider $p = 11$. What sequence does the Erdös-Turán construction return? Verify that this sequence represents location of marks on a Golomb ruler. What is the running time of this construction for arbitrary p?

Seeing through Golomb Rulers

At this point, you should have a good understanding of the mathematical concept and properties of Golomb patterns. Let's learn about an application of this concept for astrophysicists as well as Earth scientists who benefit from measurements obtained from multiple apertures. To demonstrate this application, we experiment with an image obtained from the Hubble Space Telescope as our true sky image, and see how it would be observed via employing different apertures.

Activity 1

Read and display the image tit1ed "hubble.tiff" from http://encompass.gsfc.nasa.gov/data.html. We call this image A. What are the dimensions of A?

Hint: You can use the Matlab routine called `imread` or implement its equivalent for reading .tiff files.

Activity 2

Consider matrix B with the same dimensions as A, whose cell values are initialized to zero. Place a circular aperture of radius 5 units at the center of B. Update B so that the cells representing the aperture location have value 1. Display and save grid B.

Hint 1: If B is of size $N \times N$, the center of circle should be located at $\left(\left\lfloor \frac{N}{2} \right\rfloor + 1, \left\lfloor \frac{N}{2} \right\rfloor + 1\right)$.

Hint 2: Use the definition of a circle and the location of the circle's center to determine which cells in B should have value 1.

Activity 3

Perform the below-mentioned computations to derive how image A is observed via the aperture represented in grid B. Let `fft2` and `ifft2` represent Matlab's 2D forward and inverse fast Fourier transform functions, respectively. You can use these two routines or their equivalents for calculating 2D Fourier transforms in this activity.

1. Calculate the 2D Fourier transform of A. Save the results in *2D_FFT_A*.

   Hint: Make sure the zero frequency item is at the center of the matrix by applying Matlab's `fftshift` function or its equivalent.

   2D_FFT_A = `fftshift(fft2(A))`

2. Calculate the 2D modulus square of the Fourier transform of grid B, aperture space. Call this image I. This is the *point spread function* (*PSF*) of the optical system. Computing the Fourier transform of the aperture is equal to evaluating the

Fraunhofer diffraction integral that relates the image space to aperture space.[21]

I = `fftshift(`*abs*`(fft2(B)).^2)`

3. Calculate the 2D forward Fourier transform of the PSF, I, to create image I_p. This is called the *optical transfer function* (*OTF*).[21]

Ip = *abs*`(fftshift(fft2(I)))`

4. Create the "as-seen" image in the Fourier domain by multiplying the OTF by the Fourier transform of the image using the Schur product. OTF filters the frequency content of the image (step 1). Therefore, where OTF is 0, those frequencies don't get passed by the optical system.

As_Seen_FFT = `Ip.` * *2D_FFT_A*

5. Finally, calculate the "as-seen" image in the spatial domain.

As_Seen = *abs*(`ifft2`(*As_Seen_FFT*))

6. Save the final result in a tiff file after normalizing its values to [0, 255] integer range.

How does the observed image compare to the true image in "hubble.tiff"?

Activity 4

Repeat Activities 2 and 3 with an aperture size of 7 units and 9 units. What can you conclude about the quality of the observed image and the size of the aperture used to observe it?

Activity 5

Repeat Activities 2 and 3 with three apertures and then five apertures of size 5 units that form a Golomb pattern. In other words, matrix B will contain three apertures of size 5 units in the first experiment and five apertures of size 5 units in the second experiment. How does the result compare with that of Activity 3, in which you used only one aperture? What can you conclude about the quality of the observed image and the number of apertures used to observe it?

Hint: Place aperture centers on a horizontal or vertical line such that aperture centers form a Golomb ruler and apertures do not intersect with each other.

Note: Because mission expenses are driven by aperture cost and size, Golomb aperture patterns are used to let the optical system's OTF pass more nonredundant spatial frequencies with fewer apertures.

In this case study, we learned about Golomb rulers, their properties, and applications. Golomb patterns have applications outside astrophysics and Earth sciences, however. For example, they're used in coding theory to lower information transmission rate, in crystallography to position x-ray sensors properly, and in sonar signal work.[7,12]

Golomb sequences convey something without redundant patterns, but redundancy is sometimes needed and appreciated. Redundant patterns are responsible for many beautiful art masterpieces and breath-taking musical works. Can you imagine how a song would sound without a sequence of repeated notes? Thanks to Golomb sequences, you can listen to the world's ugliest music here: http://tedxtalks.ted.com/video/TEDxMIAMI-Scott-Rickard-The-Wor! ■

Acknowledgments

I thank Dianne O'Leary and Matthew Bolcar for their careful review of this case study and helpful comments. This case study is dedicated to the memory of Richard G. Lyon (1958–2016).

References

1. S.A. Rinehart et al., "The Wide-Field Imaging Interferometry Testbed (WIIT): Recent Progress and Results," *SPIE Optical and Infrared Interferometry*, vol. 7013, 2008; http://dx.doi.org/10.1117/12.787402.
2. R.G. Lyon et al., "Wide-Field Imaging Interferometry Testbed (WIIT): Image *Construction Algorithms," SPIE Optical and Infrared Interferometry*, vol. 7013, 2008; doi: 10.1117/12.789833.
3. J.A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, 4th ed., Springer, 2005.
4. T.J. Cornwell, "A Novel Principle for Optimization of the Instantaneous Fourier Plane Coverage of Correlation Arrays," *IEEE Trans. Antennas and Propagation*, vol. 36, no. 8, 1988, pp. 1165–1167.
5. M.J.E. Golay, "Point Arrays Having Compact, Nonredundant Autocorrelations," *J. Optical Soc. Am.*, vol. 61, no. 2, 1971, pp. 272–273.
6. W.K. Klemperer, "Very Large Array Configurations for the Observation of Rapidly Varying Sources,"

Astronomy and Astrophysics Supplement, vol. 15, 1974, pp. 449–451.
7. G.S. Bloom and S.W. Golomb, "Applications of Numbered Undirected Graphs," *Proc. IEEE*, vol. 65, no. 4, 1977, pp. 562–570.
8. A. Dimitromanolakis, "Analysis of the Golomb Ruler and the Sidon Set Problems, and Determination of Large, Near-Optimal Golomb Rulers," master's thesis, Dept. Electronic and Computer Eng., Tech. Univ. Crete, June 2002.
9. S. Sidon, "Ein satz uber trigonometrische Polynome und seine Anwendungen in der Theorie der Fourier-Reihen," (in German), *Mathematische Annalen*, vol. 106, 1932, pp. 536–539.
10. J.B. Shearer, "Some New Optimum Golomb Rectangles," *Electronic J. Combinatorics*, vol. 2, 1995, article no. 12.
11. J.B. Shearer, "Symmetric Golomb Squares," *IEEE Trans. Information Theory*, vol. 50, no. 8, 2004, pp. 1846–1847.
12. S.W. Golomb and H. Taylor, "Construction and Properties of Costas Arrays," *Proc. IEEE*, vol. 72, no. 9, 1984, pp. 1143–1154.
13. K. Taylor, S. Rickard, and K. Drakakis, "Costas Arrays: Survey, Standardization, and MATLAB Toolbox," *ACM Trans. Mathematical Software*, vol. 37, no. 4, 2011, article no. 41.
14. S. Blackburn et al., "Honeycomb Arrays," *Electronic J. Combinatorics*, vol. 17, 2010; www.combinatorics.org/ojs/index.php/eljc/article/view/v17i1r172.
15. P. Hansen, B. Jaumard, and C. Meyer, "On Lower Bounds for Numbered Complete Graphs," *Discrete Applied Mathematics*, vol. 94, no. 3, 1999, pp. 205–225.
16. C. Meyer and B. Jaumard, "Equivalence of Some LP-Based Lower Bounds for the Golomb Ruler Problem," *Discrete Applied Mathematics*, vol. 154, no. 1, 2006, pp. 120–144.
17. S.W. Soliday, A. Homaifar, and G.L. Lebby, "Genetic Algorithm Approach to the Search for Golomb Rulers," *Proc. 6th Int'l Conf. Genetic Algorithms*, 1995, pp. 528–535.
18. P. Erdös, "On a Problem of Sidon in Additive Number Theory, and on Some Related Problems Addendum," J. London Mathematical Soc., vol. 19, 1944, p. 208.
19. P. Erdös and P. Turán, "On a Problem of Sidon in Additive Number Theory, and on Some Related Problems," *J. London Mathematical Soc.*, vol. 16, 1941, pp. 212–215.
20. K. Drakakis, "A Review of the Available Construction Methods for Golomb Rulers," *Advances in Mathematics of Communications*, vol. 3, no. 3, 2009, pp. 235–250.
21. J. Goodman, *Introduction to Fourier Optics*, 3rd ed., Roberts & Company Publishers, 2005.

Nargess Memarsadeghi is a senior computer engineer at NASA Goddard Space Flight Center. Her research interests include scientific computing, image processing, and optimization algorithms. Memarsadeghi received a PhD in computer science from the University of Maryland at College Park. Contact her at Nargess.Memarsadeghi@nasa.gov.